Combinatorial Generalization Bounds for Learning Ensemble of Rules

Andrey Ivahnenko (ivahnenko@forecsys.ru)
Konstantin Vorontsov (voron@forecsys.ru)

Computing Center RAS • Moscow Institute of Physics and Technology

25th European Conference on
Operational Research (EURO-XXV)
Vilnius, Lithuania • July 8 – 11, 2012

Contents

Classification problem

$\mathbb{X}^L$ — an *object space*
$f_1(x), \dots, f_n(x)$ — real-value features of an object $x \in \mathbb{X}^L$

$Y = \{1, \dots, M\}$ — a finite set of *class* labels
$y\colon X \to Y$ — unknown *target* function

$X^\ell = \{(x_1, y_1), \dots, (x_\ell, y_\ell)\}$ — *training set*, $y_i = y(x_i)$, $i = 1, \dots, \ell$

Problem: given a set X^ℓ find a classifier $r\colon X \to Y$ such that

- r is well-interpretable (humans can understand it);
- r approximates a target y on the training set X^ℓ;
- r approximates a target y everywhere on X (has a good generalization ability);

The probability of overfitting

Let $\mathbb{X}^L = \{x_1, \ldots, x_L\}$ be a finite set of objects.

Let R be a set of classifier, and $r \in R$.

Let μ be a learning method, such that $\mu(X^\ell) = \mu X^\ell = r$.

$I(r, x_i) = \bigl[r(x_i) \neq [y_i = y]\bigr]$ — binary loss function for a class y.

$\nu(r, U) = \frac{1}{|U|} \sum\limits_{x_i \in U} I(r, x_i)$ — error rate of a r on a sample U.

Assumption. All partitions $\mathbb{X}^L = X^\ell \sqcup X^k$ into an observed training set X^ℓ and a hidden testing set X^k are equiprobable.

Definition. The *probability of overfitting* is the probability that the testing error is greater that the training error by ε or more:

$$Q_\varepsilon(X^L) = \mathsf{P}\bigl[\nu(r, X^k) - \nu(r, X^\ell) \geqslant \varepsilon\bigr],$$

or

$$Q_\varepsilon(\mu, X^L) = \mathsf{P}\bigl[\nu(\mu X^\ell, X^k) - \nu(\mu X^\ell, X^\ell) \geqslant \varepsilon\bigr].$$

Vapnik-Chervonenkis bound (VC-bound), 1971

For any $\mathbb{X}^L = X^\ell \sqcup X^k$, R, μ, and $\varepsilon \in (0,1)$

$$Q_\varepsilon = \mathsf{P}\bigl[\nu(\mu X^\ell, X^k) - \nu(\mu X^\ell, X^\ell) \geqslant \varepsilon\bigr] \leqslant$$

STEP 1: *uniform bound* makes the result independent on μ:

$$\leqslant \widetilde{Q}_\varepsilon = \mathsf{P}\max_{a \in R}\bigl[\nu(r, X^k) - \nu(r, X^\ell) \geqslant \varepsilon\bigr] \leqslant$$

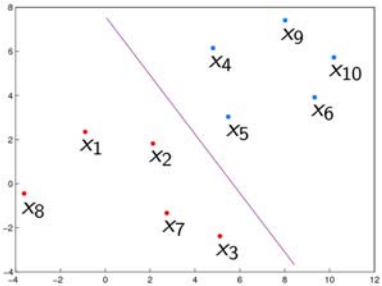
STEP 2: *union bound* (wich is usually higly overestimated):

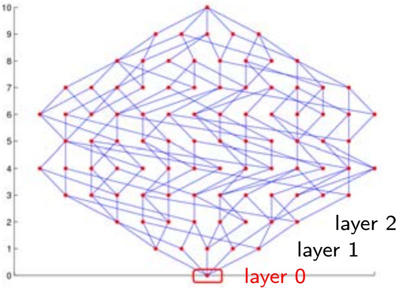
$$\leqslant \mathsf{P}\sum_{r \in R}\bigl[\nu(r, X^k) - \nu(r, X^\ell) \geqslant \varepsilon\bigr] =$$

exact one-classifier bound:

$$= \sum_{r \in R} H_L^{\ell, m}\left(\frac{\ell}{L}(m - \varepsilon k)\right), \; m = \sum_{x \in X^\ell} I(r, x)$$

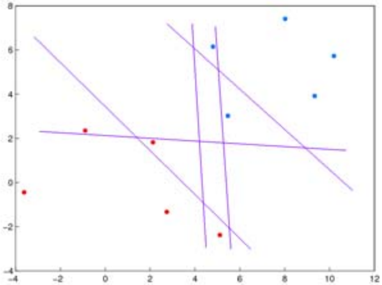
Example. Loss matrix and SC-graph for a set of linear classifiers

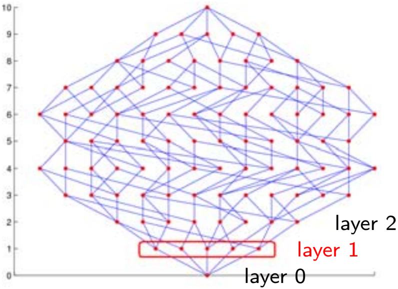




	layer 0
x_1	0
x_2	0
x_3	0
x_4	0
x_5	0
x_6	0
x_7	0
x_8	0
x_9	0
x_{10}	0

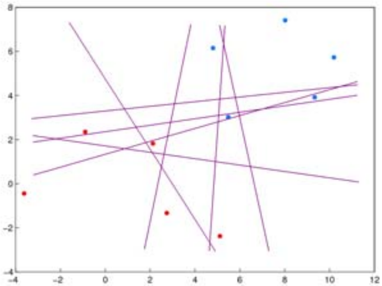
Example. Loss matrix and SC-graph for a set of linear classifiers

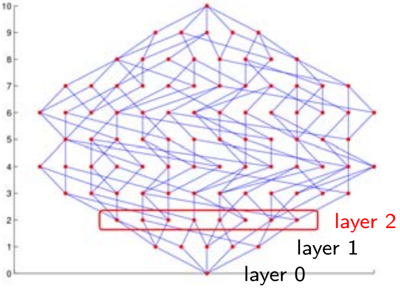




	layer 0	layer 1				
x_1	0	1	0	0	0	0
x_2	0	0	1	0	0	0
x_3	0	0	0	1	0	0
x_4	0	0	0	0	1	0
x_5	0	0	0	0	0	1
x_6	0	0	0	0	0	0
x_7	0	0	0	0	0	0
x_8	0	0	0	0	0	0
x_9	0	0	0	0	0	0
x_{10}	0	0	0	0	0	0

Example. Loss matrix and SC-graph for a set of linear classifiers





	layer 0	layer 1					layer 2								
x_1	0	1	0	0	0	0	1	0	0	0	0	1	1	0	...
x_2	0	0	1	0	0	0	1	1	0	0	0	0	0	0	...
x_3	0	0	0	1	0	0	0	1	1	0	0	0	0	1	...
x_4	0	0	0	0	1	0	0	0	1	1	0	0	0	0	...
x_5	0	0	0	0	0	1	0	0	0	1	1	1	0	0	...
x_6	0	0	0	0	0	0	0	0	0	0	1	0	1	0	...
x_7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	...
x_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
x_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
x_{10}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...

Connectivity and inferiority of a classifier

Def. *Connectivity* of a classifier $a \in A$

$p(a) = \#\{x_{ba} \in \mathbb{X}^L \colon b \prec a\}$ — low-connectivity.

$q(a) = \#\{x_{ab} \in \mathbb{X}^L \colon a \prec b\}$ — up-connectivity;

Def. *Inferiority* of a classifier $a \in A$

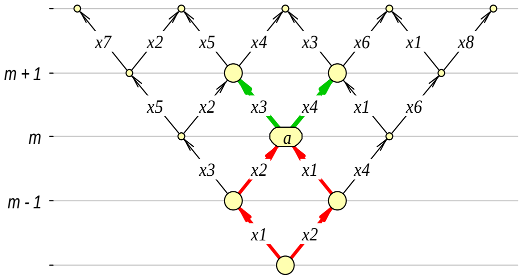
$r(a) = \#\{x_{cb} \in \mathbb{X}^L \colon c \prec b \leqslant a\} \in \{p(a), \ldots, n(a)\}$.

Example:

$p(a) = \#\{x1, x2\} = 2,$

$q(a) = \#\{x3, x4\} = 2,$

$r(a) = \#\{x1, x2\} = 2.$



The Splitting and Connectivity (SC-) bound

Theorem (SC-bound)

For any $\mathbb{X}^L$, any R and any $\varepsilon \in (0,1)$

$$Q_\varepsilon \leqslant \sum_{r \in R} \left(\frac{C_{L-q-h}^{\ell-q}}{C_L^\ell} \right) H_{L-q-h}^{\ell-q,\, m-h}\left(s_m(\varepsilon)\right),$$

where $m = L\nu(r, \mathbb{X}^L)$, $q = q(r)$, $h = h(r)$.

1. If $q(r) \equiv h(r) \equiv 0$ then SC-bound transforms to Vapnik-Chervonenkis bound: $Q_\varepsilon \leqslant \sum_{r \in R} H_L^{\ell,\, m}\left(s_m(\varepsilon)\right)$.
2. The contribution of $r \in R$ decreases exponentially by:
   $q(r) \Rightarrow$ **connected sets are less subjected to overfitting**;
   $h(r) \Rightarrow$ **only lower layers contribute significantly to Q_ε.**

Conjunctive rules

Conjunctive rule is a simple well interpretable 2-class classifier:

$$r_y(x) = \bigwedge_{j \in J} \big[f_j(x) \lessgtr_j \theta_j \big],$$

where $f_j(x)$ — features,
$J \subseteq \{1, \dots, n\}$ — subset of features, not very big, usually $|J| \lesssim 7$,
θ_j — thresholds,
$\lessgtr_j$ — one of the signs $\leqslant$ or $\geqslant$,
y — the class of the rule.

If $r_y(x) = 1$ then the rule r classifies x to the class y.

All objects x such that $r_y(x) = 0$ are not classified by r_y.

One need a lot of rules to cover all objects and build a good classifier.

Decision List and Weighted Voting of conjunctive rules

Decision list (DL) is defined by a sequence of rules $r_1(x), \dots, r_T(x)$ of respective classes $c_1, \dots, c_T \in Y$:

1: **for all** $t = 1, \dots, T$
2: **if** $r_t(x) = 1$ **then return** c_t
3: **return** c_0 *(abstain from classification)*

Weighted voting (WV) is defined by rule sets R_y of all classes $y \in Y$, with respective weights w_r for each rule r:

$$a(x) = \arg\max_{y \in Y} \sum_{r \in R_y} w_r r(x).$$

To learn DL or WV one learns rules one-by-one, gradually covering the entire training set X^ℓ (a lot of standard procedures!)

Rule evaluation metrics

The rule learning is a two-criteria optimization problem:

1) maximize the number of *positive examples* (of class y):

$$p(r_y, X^\ell) = \sum_{i=1}^{\ell} r_y(x_i)\big[y_i = y\big] \to \max_{r_y};$$

2) minimize the number of *negative examples* (not of class y):

$$n(r_y, X^\ell) = \sum_{i=1}^{\ell} r_y(x_i)\big[y_i \neq y\big] \to \min_{r_y};$$

Common practice is to combine them into one *rule evaluation metric*

$$H(p, n) \to \max_{r_y}$$

Examples of rule evaluation metrics

- Entropy criterion also called *Information gain*:
  $$h\left(\frac{P}{\ell}\right)-\frac{p+n}{\ell}h\left(\frac{p}{p+n}\right)-\frac{\ell-p-n}{\ell}h\left(\frac{P-p}{\ell-p-n}\right)\to\max,$$
  where $h(q) = -q\log_2 q - (1-q)\log_2(1-q)$;
- Gini Index — the same, but $h(q) = 2q(1-q)$;
- Fisher's exact test:
  $-\log C_P^p C_N^n / C_{P+N}^{p+n} \to \max$;
- Boosting criterion [Cohen, Singer, 1999]:
  $\sqrt{p} - \sqrt{n} \to \max$
- Meta-learning criteria [J. Fürnkranz at al., 2001–2007].

where

$$P = \bigl|\{x_i\colon y_i{=}y\}\bigr| \text{ — number of positives in the set } X^\ell;$$
$$N = \bigl|\{x_i\colon y_i{\neq}y\}\bigr| \text{ — number of negatives in the set } X^\ell.$$

The problem: rules can suffer from overfitting

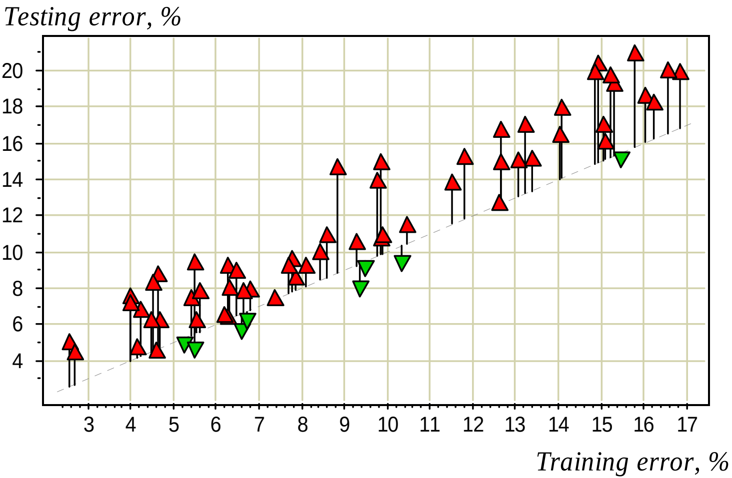
A common shortcoming of all rule evaluation metrics:

They ignore an overfitting resulting from thresholds θ_j learning.

On the independent testing set X^k

- $n(r, X^k)$ may be greater than expected;
- $p(r, X^k)$ may be less than expected.

The problem: rules are typically overfitted in real applications



Real task: predicting the result of atherosclerosis surgical treatment, $L = 98$.

SC-modification of rule evaluation metric

Problem:

Estimate $n(r, X^k)$ and $p(r, X^k)$ to select rules more carefully.

Solution:

1. Calculate data-dependent SC-bounds:

$$\mathrm{P}\left[\tfrac{1}{k}n(r, X^k) - \tfrac{1}{\ell}n(r, X^\ell) \geqslant \varepsilon\right] \leqslant \eta_n(\varepsilon);$$
$$\mathrm{P}\left[\tfrac{1}{\ell}p(r, X^\ell) - \tfrac{1}{k}p(r, X^k) \geqslant \varepsilon\right] \leqslant \eta_p(\varepsilon);$$

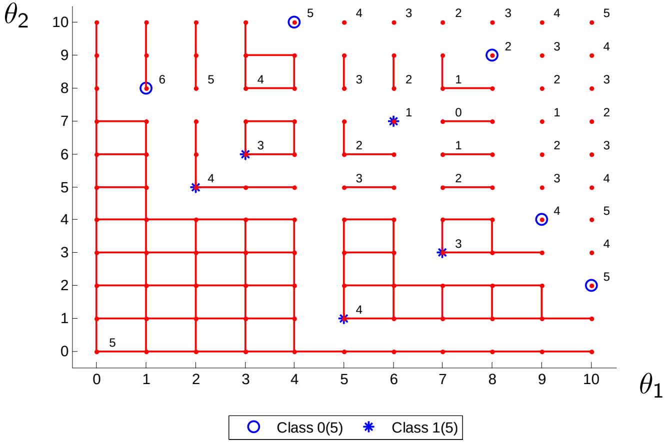
2. Invert SC-bounds: with probability at least $1 - \eta$

$$\frac{\ell}{k}n(r, X^k) \leqslant n(r, X^\ell) + \ell\varepsilon_n(\eta) \quad \equiv \hat{n}(r, X^k);$$
$$\frac{\ell}{k}p(r, X^k) \geqslant p(r, X^\ell) - \ell\varepsilon_p(\eta) \quad \equiv \hat{p}(r, X^k).$$

3. Substitute $\hat{p}$, $\hat{n}$ in evaluation metric: $H(\hat{p}, \hat{n}) \to \max\limits_r$.

Classes of equivalent rules: one point per rule

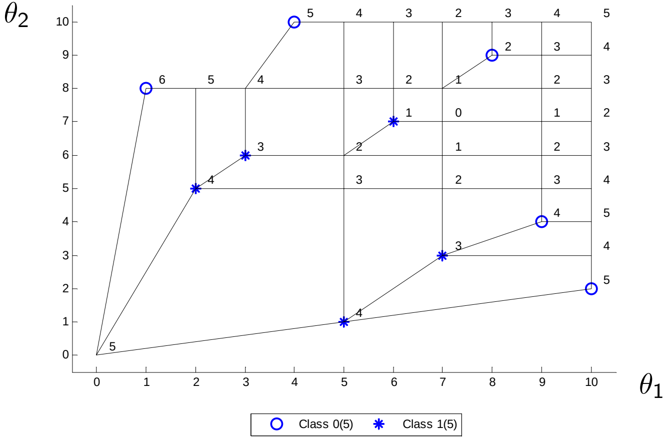
Example: separable 2-dimensional task, $L = 10$, two classes.
rules: $r(x) = \left[f_1(x) \leqslant \theta_1 \text{ and } f_2(x) \leqslant \theta_2\right]$.



Classes of equivalent rules: one point per class

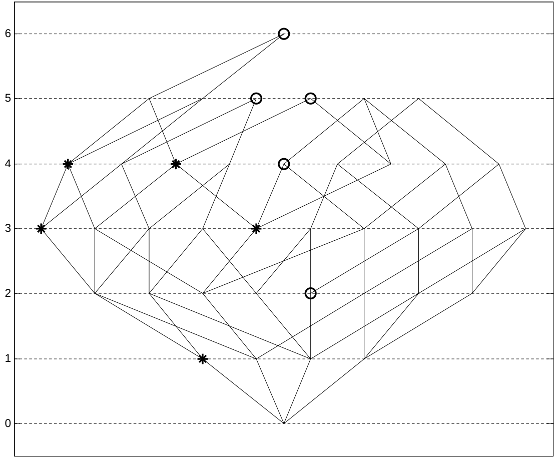
Example: the same classification task. One point per class.

rules: $r(x) = \big[f_1(x) \leqslant \theta_1 \text{ and } f_2(x) \leqslant \theta_2\big]$.



Classes of equivalent rules: SC-graph

Example: SC-graph isomorphic to the graph at previous slide.



SC-bound calculation for the set of conjunction rules

Require: features subset J, class label $y \in Y$, set of objects $\mathbb{X}^L$.
Ensure: Q_ε — SC-bound on probability of overfitting.

$R_0 :=$ the bottom rule of the SC-graph;
repeat
    for all $r \in R_0$
        find all neighbor rules $r' \in R \setminus R_0$ for the rule r;
        calculate $q := q(r)$, $h := h(r)$, $m := L\nu(r, \mathbb{X}^L)$;
        calculate the contribution of the rule r:
        $Q_\varepsilon(r) := \frac{1}{C_L^\ell} C_{L-q-h}^{\ell-q} H_{L-q-h}^{\ell-q,\, m-h}\left(\frac{\ell}{L}(m - \varepsilon k)\right)$;
        add all neighbor rules r' in R_0;
        $Q_\varepsilon := Q_\varepsilon + Q_\varepsilon(r)$;
until the contributions of layers $Q_{\varepsilon,m}$ become small.

Really, 5–10 lower layers of the SC-graph are sufficient.

Experiment on real data sets

Data sets from UCI repository:

Task	Objects	Features
australian	690	14
echo cardiogram	74	10
heart disease	294	13
hepatitis	155	19
labor relations	40	16
liver	345	6

Learning algorithms:

- WV — weighted voting (boosting);
- DL — decision list;
- LR — logistic regression.

Testing method: 10-fold cross validation.

Experiment on real data sets. Results

	tasks					
Algorithm	austr	echo	heart	hepa	labor	liver
RIPPER-opt	15.5	2.97	19.7	20.7	18.0	32.7
RIPPER+opt	15.2	5.53	20.1	23.2	18.0	31.3
C4.5(Tree)	14.2	5.51	20.8	18.8	14.7	37.7
C4.5(Rules)	15.5	6.87	20.0	18.8	14.7	37.5
C5.0	14.0	4.30	21.8	20.1	18.4	31.9
SLIPPER	15.7	4.34	19.4	17.4	12.3	32.2
LR	14.8	4.30	19.9	18.8	14.2	32.0
WV	14.9	4.37	20.1	19.0	14.0	32.3
DL	15.1	4.51	20.5	19.5	14.7	35.8
WV+CS	14.1	3.2	19.3	18.1	13.4	30.2
DL+CS	14.4	3.6	19.5	18.6	13.6	32.3

Two top results are highlighted for each task.

Conclusions

1. Splitting and connectivity properties of the set of classifiers together reduce overfitting significantly.
2. The *splitting* property:
   only a small part of classifiers are suitable for a given task.
3. The *connectivity* property:
   there a lot of similar classifiers in the set.
4. *SC-bound* is a combinatorial generalization bound that takes into account both splitting and connectivity.
5. *SC-bound* can be effectively calculated for the set of threshold conjunctive rules...
6. ...reducing the testing error by 1–2% on real data sets.

Questions, please

Andrey Ivahnenko
ivahnenko@forecsys.ru